ipu:

```
class Q {
int n;
boolean value set = false;
Synchronized int get() {
    while(valueset) {
    try {
        System.out.println("In consumer waiting \n");
        wait();
    } catch (Interrupted Exception e) {
        System.out.println("interuplend Exception caught ");
    } }
    System.out.println("Got" + n);
    value set = false;
    System.out.println("in intimate procedure");
    notify();
    return n;
    }
Synchronized void put(int n) {
    while (value set) {
        try {
        System.out.println("in Procedure waiting \n");
        wait();
        } catch(interruptedException e) {
        system.out.println("interupted Exception caught
    } }
    this.n = n;
    value set = true
    System.out.println("Put" + n);
    System.out.println("intimate Consumer");
    }
}
```

```
class Procedure implements Runnable {
Qq;
Procedure (Qq) {
  this.q = q;
  new Thread (this, "Procedure").start();
}
Public void run() {
  int i = 0;
  while (i<15) {
  q.put (i++);
  try {
    Thread.sleep (500)
  } catch (InterruptedException e) {
  System.out.println ("Procedure interrupted");
}}}}
class Consumer implements Runnable
  Qq;
  Consumer (Qq) {
    this.q = q
    new Thread (this, "Consumer").start();
  }
  Public void Run() {
    while (true) {
    q.get();
    try {
      Thread.sleep (1000);
    Procedure
    } catch (InterruptedException e) {
      System.out.println ("Consumer");
}}}}
```

```
Public class Main {
    public static void main(String[]args){
    System.out.println("Abhinav 1BM23CS008");
    Qq q=new Q();
    new producer(q);
    new consumer(q);
    }
    }.
```

Computer science 5
Information Science 5.

Q) Demonstrate inter process communication & deadlock.

Output:

Put: 1

Got: 1

Put: 2

Got: 2

Put: 3

Got: 3

Put: 4

Got: 4

Put: 5

Got: 5